IN THE CLAIMS

Please amend the claims as follows, substituting any amended claim(s) for the corresponding pending claim(s):

1. (Currently Amended) An address lookup structure comprising:

~~at least one~~ a plurality of hash tables each storing prefixes for address lookups;

a content addressable memory storing at least some prefixes for which a collision occurs within ~~the~~ at least one of the hash tables; and

a ~~block based~~ hashing lookup search mechanism that comprises:

a routing table implemented with selective hashing for a plurality of prefixes with different lengths; and

a ~~block based memory allocation unit that allocates~~ plurality of memory blocks, wherein each ~~to said at least one~~ hash table is allocated a group of the memory blocks based on a size of the respective hash table.

2. (Currently Amended) The address lookup structure according to claim 1, wherein ~~the at least one~~ each hash table is ~~contained within~~ allocated a smallest number of memory blocks sufficient to hold ~~all required~~ prefixes for which no collision occurs within the ~~at least one~~ respective hash table.

3. (Currently Amended) The address lookup structure according to claim 1, wherein ~~the at least one~~ each hash table is ~~contained within~~ allocated no more than a ~~selected limited~~ predefined number of memory blocks.

4. (Currently Amended) The address lookup structure according to claim 1, wherein ~~the at least one~~ each hash table contains prefixes hashed by one of two hash functions, a second of the two hash functions employed when a collision occurs with a first of the two hash functions.

5. (Currently Amended) The address lookup structure according to claim 1, wherein ~~the at least one hash table comprises a plurality of hash tables,~~ each hash table ~~containing~~ contains different length prefixes.

6. (Currently Amended) The address lookup structure according to claim 5, further comprising:

a ~~plurality encoded~~ priority encoder selecting a longest prefix when a plurality of matches occur between different length portions of a prefix and prefixes in each of two or more of the ~~plurality of~~ hash tables.

7. (Original) The address lookup structure according to claim 5, wherein the plurality of hash tables contain only a subset of different length prefixes possible under an addressing scheme, and wherein a remainder of the different length prefixes are stored in the content addressable memory.

8. (Currently Amended) A network router including the address lookup structure according to claim 1, the network router further comprising:

a network search engine containing the ~~at least one~~ hash tables and coupled to the content addressable memory, the network search engine performing address lookups using the ~~at least one~~ hash tables; and

an external memory coupled to the network search engine and containing per route information indexed by a next hop index generated by the network search engine.

9. (Original) A network including a plurality of interconnected network routers according to claim 8.

10. (Currently Amended) An address lookup structure comprising:

a plurality of hash tables each containing prefixes of a different length than prefixes within other hash tables within the plurality, the hash tables collectively containing only a subset of different prefix lengths less than or equal to an address length;

an additional address lookup facility handling a remainder of the different address lengths not accommodated by the plurality of hash tables; and

a ~~block-based~~ hashing lookup search mechanism that comprises:

a routing table implemented with selective hashing for a plurality of prefixes with different lengths; and

a ~~block-based memory allocation unit that allocates~~ plurality of memory blocks, wherein each ~~to said at least one~~ hash table is allocated a group of the memory blocks based on a size of the respective hash table.

11. (Original) The address lookup structure according to claim 10, wherein the additional address lookup facility comprises a content addressable memory.

12. (Previously Presented) The address lookup structure according to claim 10, wherein each of the plurality of hash tables is contained in one or more memory blocks allocated based on hashing of each prefix contained in the respective hash table using at least a first hash function,

wherein a number of memory blocks allocated to the respective hash table does not exceed a predefined number, and

wherein a remainder of prefixes of a length corresponding to prefixes within the respective hash table are handled by the additional address lookup facility.

13. (Original) The address lookup structure according to claim 10, further comprising:

a priority encoder selecting a longest prefix match from matches identified within the plurality of hash tables.

14. (Currently Amended) A method of operating an address lookup comprising:

storing at least some address prefixes in ~~at least one~~ each of a plurality of hash tables, wherein each hash table is allocated a different group of memory blocks from a plurality of memory blocks, wherein a number of the memory blocks allocated to a hash table is based on a size of the respective hash table;

storing address prefixes for which a collision occurs within ~~the~~ at least one of the hash tables in a content addressable memory; and

operating a ~~block based~~ hashing lookup search mechanism that comprises a routing table implemented with selective hashing for a plurality of prefixes with different lengths~~ and a block based memory allocation unit that allocates memory blocks to said at least one hash table~~.

15. (Currently Amended) The method according to claim 14, further comprising:

maintaining ~~the at least one~~ each hash table within a smallest number of memory blocks sufficient to hold all required prefixes for which no collision occurs within the ~~at least one~~ respective hash table.

16. (Currently Amended) The method according to claim 14, further comprising:

~~maintaining the at least one~~ allocating each hash table ~~within~~ no more than a ~~selected limited~~ predefined number of memory blocks.

17. (Currently Amended) The method according to claim 14, further comprising:

hashing prefixes in ~~the at least one~~ each hash table with one of two hash functions, a second of the two hash functions employed when a collision occurs with a first of two hash functions.

18. (Original) The method according to claim 14, further comprising:

storing, in each of a plurality of hash tables, prefixes of a different length than prefixes contained in any other of the plurality of hash tables.

19. (Original) The method according to claim 18, further comprising:

selecting a longest prefix when a plurality of matches occur between different length portions of a prefix and prefixes in each of two or more of the plurality of hash tables.

20. (Previously Presented) The method according to claim 18, further comprising:

storing prefixes corresponding to only a subset of different prefix lengths possible under an addressing scheme in the plurality of hash tables; and

storing a remainder of prefixes in the content addressable memory.

Please add the following new claim(s):

21. (Newly Added) The address lookup structure according to claim 1, wherein the hashing lookup search mechanism further comprises:

a configuration register associated with each memory block, each configuration register identifying the prefix length to which the respective memory block is allocated.

21. (Newly Added) The address lookup structure according to claim 1, wherein each configuration register identifies the hash function to which the respective memory block is allocated.